

# Number Balancing is as hard as Minkowski's Theorem and Shortest Vector

Rebecca Hoberg\*   Harishchandra Ramadas†   Thomas Rothvoss‡   Xin Yang§

University of Washington, Seattle, WA 98195

## Abstract

The number balancing (NBP) problem is the following: given real numbers  $a_1, \dots, a_n \in [0, 1]$ , find two disjoint subsets  $I_1, I_2 \subseteq [n]$  so that the difference  $|\sum_{i \in I_1} a_i - \sum_{i \in I_2} a_i|$  of their sums is minimized. An application of the pigeonhole principle shows that there is always a solution where the difference is at most  $O(\frac{\sqrt{n}}{2^n})$ . Finding the minimum, however, is NP-hard. In polynomial time, the *differencing algorithm* by Karmarkar and Karp from 1982 can produce a solution with difference at most  $n^{-\Theta(\log n)}$ , but no further improvement has been made since then.

In this paper, we show a relationship between NBP and Minkowski's Theorem. First we show that an approximate oracle for Minkowski's Theorem gives an approximate NBP oracle. Perhaps more surprisingly, we show that an approximate NBP oracle gives an approximate Minkowski oracle. In particular, we prove that any polynomial time algorithm that guarantees a solution of difference at most  $2^{\sqrt{n}}/2^n$  would give a polynomial approximation for Minkowski as well as a polynomial factor approximation algorithm for the Shortest Vector Problem.

## 1 Introduction

One of *six basic NP-complete problems* of Garey and Johnson [GJ97] is the *partition problem* that for a list of numbers  $a_1, \dots, a_n$  asks whether there is a partition of the indices so that the sums of the numbers in both partitions coincide. Partition and related problems like knapsack, subset sum and bin packing are some of the fundamental classical problems in theoretical computer science with numerous practical applications; see for example the textbooks [MT90, KPP04] and the article of Mertens [Mer06]. In this paper, we study a variant called the *number balancing problem* (NBP), where the goal is to find two disjoint subsets  $I_1, I_2 \subseteq \{1, \dots, n\}$  so that the difference  $|\sum_{i \in I_1} a_i - \sum_{i \in I_2} a_i|$  is minimized. Equivalently, given a vector of numbers  $\mathbf{a} = (a_1, \dots, a_n) \in [0, 1]^n$ , we want to find a vector of *signs*  $\mathbf{x} \in \{-1, 0, 1\}^n \setminus \{\mathbf{0}\}$  so that  $|\langle \mathbf{a}, \mathbf{x} \rangle| = |\sum_{i=1}^n x_i a_i|$  is minimized. Woeginger and Yu [WY92] studied this problem under the name “equal-subset-sum” and showed that it is NP-hard to decide whether there are two disjoint subsets that sum up to the exact same value. This version has also been extensively studied in combinatorics [Lun88, Boh96, LY11].

On the positive side, it is not hard to prove that there is always a solution with exponentially small error. Suppose that  $a_1, \dots, a_n \in [0, 1]$ . Consider the list of  $2^n$  many numbers  $\sum_{i=1}^n a_i x_i$  for all  $\mathbf{x} \in \{0, 1\}^n$ . All these numbers fall into the interval  $[0, n]$ , hence by the *pigeonhole principle*, we can find two distinct vectors  $\mathbf{x}, \mathbf{x}' \in \{0, 1\}^n$  with  $|\sum_{i=1}^n a_i x_i - \sum_{i=1}^n a_i x'_i| \leq \frac{n}{2^n - 1}$ . Then  $\mathbf{x} - \mathbf{x}'$  gives the desired solution. Note that the bound can be slightly improved to  $O(\frac{\sqrt{n}}{2^n})$  by using the fact that due to *concentration of measure* effects, for

---

\*Email: rahoberg@uw.edu

†Email: ramadas@uw.edu

‡Email: rothvoss@uw.edu. Supported by NSF grant 1420180 with title “*Limitations of convex relaxations in combinatorial optimization*”, an Alfred P. Sloan Research Fellowship and a David & Lucile Packard Foundation Fellowship.

§Email: yx1992@uw.edu

a constant fraction of vectors  $\mathbf{x} \in \{0, 1\}^n$ , the sums  $\sum_{i=1}^n a_i x_i$  fall into an interval of length  $\sqrt{n}$  (instead of  $n$ ).

However, since these arguments rely on the pigeonhole principle, they are non-constructive. Restricting the non-constructive argument to polynomially many “pigeons” provides a simple polynomial time algorithm to find at least an  $\mathbf{x} \in \{-1, 0, 1\}^n \setminus \{\mathbf{0}\}$  with  $|\langle \mathbf{a}, \mathbf{x} \rangle| \leq \frac{1}{\text{poly}(n)}$  for an arbitrarily small polynomial. Interestingly, the only known polynomial time algorithm that gives a better guarantee is Karmarkar and Karp’s *differencing algorithm* [KK82] which provides the bound  $|\langle \mathbf{a}, \mathbf{x} \rangle| \leq n^{-c \log(n)}$  for some constant  $c > 0$ . Their algorithm uses a recursive scheme; find  $\Theta(n)$  pairs of numbers  $a_i$  of distance at most  $\Theta(\frac{1}{n})$  and create an instance consisting of their differences, then recurse.

This leads to the natural question: *Given  $a_1, \dots, a_n \in [0, 1]$ , what upper bound on  $|\sum_{i=1}^n a_i x_i|$  can be guaranteed if  $\mathbf{x} \in \{-1, 0, 1\}^n \setminus \{\mathbf{0}\}$  is to be chosen in polynomial time?*

While answering this question directly seems out of reach, it appears that NBP falls into a class of problems where good solutions exist due to the pigeonhole principle, such as the *Shortest Vector Problem* or *Minkowski’s Theorem*. Recall that given linearly independent vectors  $\mathbf{b}_1, \dots, \mathbf{b}_n \in \mathbb{R}^n$ , a (full rank) lattice is the set  $\Lambda := \{\sum_{i=1}^n \lambda_i \mathbf{b}_i : \lambda_i \in \mathbb{Z} \ \forall i = 1, \dots, n\}$ . The set  $\{\mathbf{b}_1, \dots, \mathbf{b}_n\}$  is called a *basis* for  $\Lambda$  and we define  $\det(\Lambda) := |\det(B)|$ . The Shortest vector problem then consists of finding a non-zero vector in  $\Lambda$  that minimizes the Euclidean norm. The famous LLL-algorithm [LLL82] can find a  $2^{n/2}$ -approximation in polynomial time (the generalized *block reduction method* of Schnorr [Sch87] brings the factor down to  $2^{n \log \log(n) / \log(n)}$ ). As a rarity in theoretical computer science, the shortest vector problem admits (NP  $\cap$  coNP)-certificates for a value that is at most a factor  $O(\sqrt{n})$  away from the optimum [AR05], while the best known hardness lies at a subpolynomial bound of  $n^{\Theta(1/\log \log n)}$  [HR07]. Using the pigeonhole principle one can show that a lattice  $\Lambda$  contains a vector of length at most  $O(\sqrt{n}) \cdot \det(\Lambda)^{1/n}$ . Interestingly, a polynomial time algorithm that achieves this bound would imply an  $O(n)$ -approximation algorithm even for worst-case lattices [Ajt96], enough to break lattice-based cryptosystems [LLJS90].

*Minkowski’s Theorem* tells us that any symmetric convex body  $K \subseteq \mathbb{R}^n$  of volume at least  $2^n$  must intersect  $\mathbb{Z}^n \setminus \{\mathbf{0}\}$ , see for example [Mat02]. This theorem is proven by placing translates of  $\frac{1}{2}K$  at any lattice point and then inferring an overlap due to the pigeonhole principle. Again, one can consider the algorithmic question: *given a symmetric convex body  $K$  with volume at least  $2^n$ , for what factor  $\rho$  can one be guaranteed to find an  $\mathbf{x} \in (\rho K) \cap (\mathbb{Z}^n \setminus \{\mathbf{0}\})$  in polynomial time?* We would like to point out that this factor  $\rho$  is within a polynomial factor of the shortest vector approximability using the fact that there is a linear transformation sandwiching  $K$  between two Euclidean balls whose radius differs by a factor of  $\sqrt{n}$  [Joh48].

It is not hard to use an *exact* oracle for Minkowski’s Theorem to find a good number balancing solution, since the body  $K := \{\mathbf{x} \in (-2, 2) : |\sum_{i=1}^n x_i a_i| \leq \Theta(\frac{n}{2^n})\}$  has a volume of  $2^n$ . However, it is not clear how we could use an *approximate* oracle. For example, it is known that the LLL-algorithm can be used to find a nonzero integer vector  $\mathbf{x} \in \rho K$  for a factor of  $\rho = \text{poly}(n) \cdot 2^{n/2}$ . While the error guarantee of  $|\sum_{i=1}^n a_i x_i| \leq \text{poly}(n) \cdot 2^{-n/2}$  outperforms the Karmarkar-Karp algorithm, we only know that  $\|\mathbf{x}\|_\infty < 2\rho$ , which means that  $\mathbf{x}$  will not be a valid solution if  $\rho > 1^1$ . This leads us to the next question: *what factor  $\rho$  is needed for Minkowski’s Theorem to improve over Karmarkar-Karp’s bound?* Again, we should mention that a  $\text{poly}(n)$ -approximation for Minkowski is not known to be inconsistent with NP  $\neq$  P.

We have seen that in a certain sense the Shortest Vector Problem and Minkowski’s Theorem are generalizations of number balancing. This brings us to the question about the reverse: *given an oracle that solves NBP within an exponentially small error, can this give a non-trivial oracle for the Shortest Vector Problem or Minkowski’s Theorem?*

## 1.1 Contribution

In this work, we provide some answers to the questions raised above, by relating the complexity of the number balancing problem to Minkowski’s Theorem. For  $\rho \geq 1$ , we define a  $\rho$ -approximation for the Minkowski problem as a polynomial time algorithm that on input<sup>2</sup> of a symmetric convex set  $K \subseteq \mathbb{R}^n$  with  $\text{vol}_n(K) > 2^n$ ,

<sup>1</sup>If  $\rho \leq 2 - \varepsilon$ , then one can still obtain an error of  $|\sum_{i=1}^n a_i x_i| \leq 2^{-\Theta(\varepsilon n)}$ , but this breaks down if  $\rho \geq 2$ .

<sup>2</sup>We will specify in the text, how the body  $K$  is represented.

finds a vector in  $(\rho \cdot K) \cap (\mathbb{Z}^n \setminus \{\mathbf{0}\})$ . Moreover, for  $\delta > 0$  we define a  $\delta$ -approximation for the number partitioning problem as a polynomial time algorithm that receives  $\mathbf{a} \in [0, 1]^n$  as input and produces a vector  $\mathbf{x} \in \{-1, 0, 1\}^n \setminus \{\mathbf{0}\}$  with  $|\langle \mathbf{a}, \mathbf{x} \rangle| \leq \delta$ . We provide the following reduction:

**Theorem 1.** *Suppose there is a  $\rho$ -approximation for Minkowski's problem for polytopes  $K$  with  $O(n)$  facets. Then there is a  $\delta$ -approximation for number balancing where  $\delta := 2^{-n^{\Theta(1/\rho)}}$*

In fact, it suffices to have such an oracle for the linear transformation of a cube, which is equivalent to an oracle for SVP in the  $\|\cdot\|_\infty$  norm.

**Theorem 2.** *Suppose that there is a polynomial time algorithm that given a lattice  $\Lambda \subseteq \mathbb{R}^n$  with  $\det(\Lambda) \leq 1$  finds a non-zero vector  $\mathbf{x} \in \Lambda$  of length  $\|\mathbf{x}\|_\infty \leq \rho$ . Then there is  $\delta$ -approximation for number balancing where  $\delta := 2^{-n^{\Theta(1/\rho)}}$ .*

Recall that  $\text{vol}_n([-1, 1]^n) = 2^n$  and hence for such a lattice there is always an  $\mathbf{x} \in \Lambda \setminus \{\mathbf{0}\}$  with  $\|\mathbf{x}\|_\infty \leq 1$ . In particular an oracle for  $\rho \leq c' \log(n)/\log \log(n)$  would imply an improvement over Karmarkar-Karp's algorithm, where  $c' > 0$  is a small enough constant. Again, we would like to stress that the NP-hardness bounds of [HR07] for Shortest Vector do not apply for such lattices where a solution is guaranteed to exist.

Finally, we can also prove that an oracle with exponentially small error for number balancing would provide approximations for Minkowski's Theorem and Shortest Vector:

**Theorem 3.** *Suppose that there is a  $\delta$ -approximation for number balancing with  $\delta \leq 2^{\sqrt{n}}/2^n$ . Then there is an  $O(n^5)$ -approximation for Minkowski's problem. Here it suffices to have a separation oracle for the convex body  $K \subseteq \mathbb{R}^n$ .*

## 2 Reducing Number Balancing to Minkowski's Theorem

In this section we will show how to solve NBP with an oracle for Minkowski's Theorem. The idea is to consider a hypercube intersected with the constraint  $|\langle \mathbf{a}, \mathbf{x} \rangle| \leq \delta$ , and to show that this set has large enough volume. If we have an exact Minkowski oracle, this gives us  $\mathbf{x} \in \{-1, 0, 1\}^n \setminus \{\mathbf{0}\}$  as desired. Here we state a more general version which uses only a  $\rho$ -approximate Minkowski oracle, and then show how we can use this more general version to solve NBP with a weaker bound.

**Theorem 4.** *Suppose we have a  $\rho$ -approximate Minkowski oracle, and let  $k > 0$  be any positive integer. Then, for any  $\mathbf{a} \in [0, 1]^n$ , there is a polynomial time algorithm to find  $\mathbf{x} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$  with  $\|\mathbf{x}\|_\infty \leq k$  and so that  $|\langle \mathbf{a}, \mathbf{x} \rangle| \leq n \left( \frac{\rho}{k+1} \right)^{n-1}$ .*

*Proof.* For ease of notation, let

$$\delta := n \left( \frac{\rho}{k+1} \right)^{n-1}.$$

Now consider the body

$$K := \left\{ \mathbf{x} \in \left( -\frac{k+1}{\rho}, \frac{k+1}{\rho} \right)^n : |\langle \mathbf{a}, \mathbf{x} \rangle| \leq \delta \right\}.$$

Obviously this is a symmetric convex body. For  $\alpha \in [-\delta, \delta]$ , consider the  $(n-1)$ -dimensional slice

$$K(\alpha) := \left\{ \mathbf{x} \in \left( -\frac{k+1}{\rho}, \frac{k+1}{\rho} \right)^n : \langle \mathbf{a}, \mathbf{x} \rangle = \alpha \right\}$$

of it. Let us consider the  $(n-1)$ -dimensional volume  $\text{vol}_{n-1}(K(\alpha))$  of that slice. Then we can write the volume of  $K$  as

$$\text{vol}_n(K) = \int_{-\delta}^{\delta} \text{vol}_{n-1}(K(\alpha)) d\alpha.$$

Moreover

$$\left(\frac{2(k+1)}{\rho}\right)^n = \text{vol}_n\left(-\frac{k+1}{\rho}, \frac{k+1}{\rho}\right)^n = \int_{-\frac{n(k+1)}{\rho}}^{\frac{n(k+1)}{\rho}} \text{vol}_{n-1}(K(\alpha)) d\alpha.$$

since the slices are empty if  $|\alpha| > \frac{n(k+1)}{\rho}$ . By symmetry  $\text{vol}_{n-1}(K(\alpha)) = \text{vol}_{n-1}(K(-\alpha))$ . Moreover, by convexity of  $K$ , for  $\alpha \geq 0$ , the quantity  $\text{vol}_{n-1}(K(\alpha))$  is monotonically non-increasing. Thus

$$\text{vol}_n(K) = \int_{-\delta}^{\delta} \text{vol}_{n-1}(K(\alpha)) d\alpha \geq \frac{\delta}{\left(\frac{n(k+1)}{\rho}\right)} \cdot \int_{-\frac{n(k+1)}{\rho}}^{\frac{n(k+1)}{\rho}} \text{vol}_{n-1}(K(\alpha)) d\alpha = \frac{\delta}{\left(\frac{n(k+1)}{\rho}\right)} \cdot \left(\frac{2(k+1)}{\rho}\right)^n.$$

Now, using the fact that  $\delta = n\left(\frac{\rho}{k+1}\right)^{n-1}$ , we get  $\text{vol}_n(K) \geq 2^n$ . Hence, using our  $\rho$ -approximate Minkowski oracle, we can find a vector

$$\mathbf{x} \in (\rho K \cap \mathbb{Z}^n) \setminus \{\mathbf{0}\} = ((-(k+1), (k+1))^n \cap \mathbb{Z}^n) \setminus \{\mathbf{0}\}.$$

In particular, this gives  $\mathbf{x} \in \mathbb{Z}^n$  with  $|\langle \mathbf{a}, \mathbf{x} \rangle| \leq n\left(\frac{\rho}{k+1}\right)^{n-1}$  and  $\|\mathbf{x}\|_{\infty} \leq k$ .  $\square$

The bound in Theorem 4 can be strengthened by a  $\sqrt{n}$  factor by using concentration of measure arguments. We omit this here.

Now suppose, for instance, that we have a  $(2 - \epsilon)$ -approximate oracle for some  $\epsilon \in (0, 1]$ . Then we can pick  $k = 1$  and get

$$|\langle \mathbf{a}, \mathbf{x} \rangle| \leq n\left(\frac{2 - \epsilon}{2}\right)^{n-1} \leq n \cdot \exp\left(-\frac{\epsilon(n-1)}{2}\right) = 2^{-\Theta(\epsilon n)},$$

with  $\|\mathbf{x}\|_{\infty} \leq 1$  and  $\mathbf{x} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$ . However, this line of arguments breaks down if we only have access to a  $\rho$ -approximation for  $\rho \geq 2$  as these would in general not produce feasible solutions for number balancing.

It turns out that we can design a *recursive self-reduction*. The main technical argument is to transform an algorithm that finds  $\mathbf{x} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$  with  $\|\mathbf{x}\|_{\infty} \leq k$ , into an algorithm that finds vectors  $\mathbf{x} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$  with  $\|\mathbf{x}\|_{\infty} \leq \frac{k}{2}$ , with a bounded decay in the error  $|\langle \mathbf{a}, \mathbf{x} \rangle|$ . Applying this recursively gives the following lemma.

**Lemma 5.** *Suppose that there is a polynomial-time algorithm that for any  $\mathbf{a}' \in [-1, 1]^n$  finds a vector  $\mathbf{x}' \in \{-k, \dots, k\}^n \setminus \{\mathbf{0}\}$  with  $|\langle \mathbf{a}', \mathbf{x}' \rangle| \leq 2^{-n}$ . If  $k \leq \frac{\log n}{6 \log \log n}$ , then there is also a polynomial time algorithm that for any  $\mathbf{a} \in [-1, 1]^n$  finds a vector  $\mathbf{x} \in \mathbb{Z}^n$  with  $|\langle \mathbf{a}, \mathbf{x} \rangle| \leq 2^{-n^{\frac{1}{3k}}}$  and  $\mathbf{x} \in \{-1, 0, 1\}^n \setminus \{\mathbf{0}\}$ .*

The way we do the self-reduction is the following. We partition our set of  $n$  numbers into subsets of size  $\sqrt{n}$ . First, for each subset  $\ell$ , we find a number  $b_{\ell} \neq 0$  for which we can (approximately) express  $b_{\ell}, 2b_{\ell}, \dots, kb_{\ell}$  as linear combinations of elements of that subset using only coefficients in  $\{-\lfloor \frac{k}{2} \rfloor, \dots, \lfloor \frac{k}{2} \rfloor\}$ . We then run our assumed algorithm on  $b_1, \dots, b_{\sqrt{n}}$  to obtain  $\mathbf{y} \in \{-k, \dots, k\}^n$  with  $\langle \mathbf{b}, \mathbf{y} \rangle = \sum_{\ell=1}^{\sqrt{n}} y_{\ell} b_{\ell}$  being small. Since each of the summands can be expressed more efficiently in terms of our original set of numbers, we obtain a good solution  $\mathbf{x}$  with coefficients in  $\{-\lfloor \frac{k}{2} \rfloor, \dots, \lfloor \frac{k}{2} \rfloor\}$ .

The following two lemmas go through this argument more precisely. The first gives a condition under which we can find a more efficient coefficient representation for a number  $\beta$ . In the second, we will show that an application of the assumed algorithm will actually allow us to satisfy this condition, and so we are able to construct the stronger oracle. Note that the interesting parameter choice is  $r := \lceil k/2 \rceil$ , so that the size of the coefficients is halved.

**Lemma 6.** *Let  $r, k \in \mathbb{N}$  be parameters with  $r < k$ . Let  $\alpha_1, \dots, \alpha_k \in \mathbb{R}$  so that  $\sum_{i=1}^k i \cdot \alpha_i = 0$  and abbreviate  $\beta := \alpha_r + \dots + \alpha_k$ . Then  $\{j \cdot \beta \mid j = -k, \dots, k\} \subseteq \{\sum_{i=1}^k \lambda_i \alpha_i \mid \lambda_i \in \mathbb{Z} \text{ and } |\lambda_i| \leq \max\{r-1, k-r\} \text{ for } i = 1, \dots, k\}$ .*

*Proof.* By symmetry it suffices to consider  $j \geq 0$ . For  $j \in \{0, \dots, r-1\}$ , we can obviously write

$$j \cdot \beta = j \cdot \alpha_r + \dots + j \cdot \alpha_k.$$

Now consider  $j \in \{r, \dots, k\}$ . The trick is to use that

$$r \cdot \beta + \sum_{i=1}^{r-1} i \cdot \alpha_i + \sum_{i=r}^k (i-r) \cdot \alpha_i = \sum_{i=1}^k i \cdot \alpha_i = 0 \quad \Rightarrow \quad r \cdot \beta = \sum_{i=1}^{r-1} (-i) \cdot \alpha_i + \sum_{i=r}^k (r-i) \cdot \alpha_i.$$

Then

$$j \cdot \beta = (j-r) \cdot \beta + r \cdot \beta = \sum_{i=1}^{r-1} (-i) \cdot \alpha_i + \sum_{i=r}^k \underbrace{[(j-r) + (r-i)]}_{=j-i} \cdot \alpha_i$$

If we inspect the size of the used coefficients, then for  $i \in \{1, \dots, r-1\}$  we have  $|-i| \leq r-1$  and for  $i \in \{r, \dots, k\}$  we have  $|j-i| \leq k-r$ .  $\square$

Note that if we have the weaker assumption of  $|\sum_{i=1}^k i \cdot \alpha_i| \leq \delta$ , then for any  $j \in \{0, \dots, k\}$  one can find coefficients  $\lambda_i$  with  $|j \cdot \beta - \sum_{i=1}^k \lambda_i \alpha_i| \leq \delta$ .

**Lemma 7.** Let  $k, r \in \mathbb{N}$  be parameters with  $0 < r < k$ . Let  $f : \mathbb{N} \rightarrow \mathbb{R}$  be a non-negative function such that  $f(n) \geq 4 \log n$ . Suppose that there is a polynomial-time algorithm that for any  $\mathbf{a}' \in [-1, 1]^n$  finds a vector  $\mathbf{x}' \in \{-k, \dots, k\}^n \setminus \{\mathbf{0}\}$  with  $|\langle \mathbf{a}', \mathbf{x}' \rangle| \leq 2^{-f(n)}$ . Then there is also a polynomial time algorithm that for any  $\mathbf{a} \in [-1, 1]^n$  finds a vector  $\mathbf{x} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$  with  $|\langle \mathbf{a}, \mathbf{x} \rangle| \leq 2^{-f(\sqrt{n})/2}$  and  $\|\mathbf{x}\|_\infty \leq \max\{r-1, k-r\}$ .

*Proof.* Let  $\mathbf{a} \in [-1, 1]^n$  be the given vector of numbers. Split  $[n]$  into blocks  $I_1, \dots, I_{\sqrt{n}}$  each of size  $|I_\ell| = \sqrt{n}$ . For each block  $I_\ell$  we use the oracle to find a vector  $\mathbf{x}_\ell \in \{-k, \dots, k\}^n \setminus \{\mathbf{0}\}$  with  $\text{supp}(\mathbf{x}_\ell) \subseteq I_\ell$  so that  $|\langle \mathbf{a}, \mathbf{x}_\ell \rangle| \leq 2^{-f(\sqrt{n})}$ . If for any  $\ell$  one has  $\|\mathbf{x}_\ell\|_\infty \leq r-1$ , then we simply return  $\mathbf{x} := \mathbf{x}_\ell$  and are done. Otherwise, we write the vector as  $\mathbf{x}_\ell = \sum_{i=1}^k i \cdot \mathbf{x}_{\ell,i}$  with vectors  $\mathbf{x}_{\ell,1}, \dots, \mathbf{x}_{\ell,k} \in \{-1, 0, 1\}^n$ . Note that these vectors will have disjoint support and  $\text{supp}(\mathbf{x}_{\ell,1}), \dots, \text{supp}(\mathbf{x}_{\ell,k}) \subseteq I_\ell$ . Moreover we know that for every  $\ell$  there is at least one index  $i \in \{r, \dots, k\}$  with  $\mathbf{x}_{\ell,i} \neq \mathbf{0}$ .

Now define a vector  $\mathbf{b} \in \mathbb{R}^{\sqrt{n}}$  with  $b_\ell := \sum_{i=r}^k \langle \mathbf{a}, \mathbf{x}_{\ell,i} \rangle$ . Note that if for any  $\ell$  we have  $|b_\ell| \leq 2^{-f(\sqrt{n})}$ , then we can set  $\mathbf{x} = \sum_{i=r}^k \mathbf{x}_{\ell,i}$  and we are done. Therefore we may assume that  $|b_\ell| > 2^{-f(\sqrt{n})}$  for all  $\ell$ . Also note that  $\|\mathbf{b}\|_\infty \leq \sqrt{n}$ . We run the oracle again to find a vector  $\mathbf{y} \in \{-k, \dots, k\}^{\sqrt{n}} \setminus \{\mathbf{0}\}$  so that  $|\langle \mathbf{b}, \mathbf{y} \rangle| \leq \sqrt{n} \cdot 2^{-f(\sqrt{n})}$ . For each block  $\ell \in [\sqrt{n}]$  we can use Lemma 6 to find integer coefficients  $\lambda_{\ell,i}$  with  $|\lambda_{\ell,i}| \leq \max\{r-1, k-r\}$  so that

$$\left| y_\ell \cdot b_\ell - \sum_{i=1}^k \lambda_{\ell,i} \cdot \langle \mathbf{a}, \mathbf{x}_{\ell,i} \rangle \right| \leq 2^{-f(\sqrt{n})}.$$

We define

$$\mathbf{x} := \sum_{\ell=1}^{\sqrt{n}} \sum_{i=1}^k \lambda_{\ell,i} \mathbf{x}_{\ell,i}.$$

Then  $\|\mathbf{x}\|_\infty \leq \max\{r-1, k-r\}$  since the  $\mathbf{x}_{\ell,i}$ 's have disjoint support and  $\|\mathbf{x}_{\ell,i}\|_\infty \leq 1$  for all  $\ell, i$ . Moreover, since there is some  $y_\ell \neq 0$  and  $|b_\ell| > 2^{-f(\sqrt{n})}$ , we have  $\mathbf{x} \neq \mathbf{0}$ .

Finally we inspect that

$$|\langle \mathbf{a}, \mathbf{x} \rangle| \leq |\langle \mathbf{y}, \mathbf{b} \rangle| + \sum_{\ell=1}^{\sqrt{n}} \left| y_\ell b_\ell - \sum_{i=1}^k \lambda_{\ell,i} \langle \mathbf{a}, \mathbf{x}_{\ell,i} \rangle \right| \leq 2\sqrt{n} \cdot 2^{-f(\sqrt{n})} \leq 2^{-f(\sqrt{n})/2}.$$

The last line comes from the fact that when  $f(n) \geq 4 \log n$ , we have  $2\sqrt{n} \leq 2^{\frac{2}{3} \log n} = 2^{\frac{4}{3} \log \sqrt{n}} \leq 2^{\frac{1}{3} f(\sqrt{n})}$ .  $\square$

Now we can apply Lemma 7 recursively to prove Lemma 5.

*Proof of Lemma 5.* For  $k \in \mathbb{N}$ , take  $r = \lfloor k/2 \rfloor$ , then Lemma 7 says any  $\mathbf{a} \in [-1, 1]^n$ , if we have a  $\mathbf{x} \in \{-k, \dots, k\}^n$  such that  $|\langle \mathbf{a}, \mathbf{x} \rangle| \leq 2^{-f(n)}$ , we can obtain  $\mathbf{x}' \in \{-\lfloor k/2 \rfloor, \dots, \lfloor k/2 \rfloor\}^n$  such that  $|\langle \mathbf{a}, \mathbf{x}' \rangle| \leq 2^{-f(\sqrt{n})/2}$ . Repeat this procedure for  $t := \lceil \log k \rceil$  times, if  $f(n^{2^{-t}}) \geq 4 \log n$ , we will get  $\mathbf{x}'' \in \{-1, 0, 1\}^n$  such that  $|\langle \mathbf{a}, \mathbf{x}'' \rangle| \leq 2^{-f(n^{2^{-t}})/2^t}$ .

So we can take  $f(n) = n$  as appearing in Lemma 7. Note that when  $k \leq \frac{\log n}{6 \log \log n}$ ,  $t = \lceil \log k \rceil \leq \log(\frac{1}{2} \frac{\log n}{\log \log n})$ , so

$$f(n^{2^{-t}}) = n^{2^{-t}} \geq n^{2^{-\log(\frac{1}{2} \frac{\log n}{\log \log n})}} = \log^2 n \geq 4 \log n.$$

Then the condition in Lemma 7 is satisfied, and the bound we get is

$$2^{-f(n^{2^{-t}})/2^t} = 2^{-(n^{2^{-t}})/2^t} \leq 2^{-\frac{1}{2k}} \leq 2^{-n^{\frac{1}{3k}}}.$$

Here we use the fact that  $t = \lceil \log k \rceil \leq \log 2k$ , and when  $k \leq \frac{1}{6} \frac{\log n}{\log \log n}$ , we have  $\frac{n^{\frac{1}{2k}}}{2k} \geq n^{\frac{1}{3k}}$ .  $\square$

Using Lemma 5, we are now able to prove Theorem 1:

*Proof of Theorem 1.* If  $\rho \geq \frac{\log n}{48 \log \log n}$ , then  $2^{-n^{\Theta(1/\rho)}} = 2^{-\log^{O(1)} n}$ . By choosing a proper constant on the exponent, this can be achieved with the Karmarkar-Karp algorithm. So we only need to work with  $\rho < \frac{\log n}{48 \log \log n}$ .

If we take  $k = 3\rho$ , then Theorem 4 implies  $|\langle \mathbf{a}, \mathbf{x} \rangle| \leq n \left( \frac{\rho}{k+1} \right)^{n-1} \leq 2^{-n}$ . Moreover,  $k = 3\rho \leq \frac{\log n}{16 \log \log n}$ , and hence the condition of Lemma 5 is satisfied. Then the bound given by Lemma 5 is  $2^{-n^{\frac{1}{3k}}} \leq 2^{-n^{\Theta(1/\rho)}}$ .  $\square$

Instead of using an oracle for Minkowski's Theorem one can directly use an oracle for the Shortest Vector Problem in the  $\|\cdot\|_\infty$  norm. We need the following theorem:

**Theorem 8.** Suppose that there is a polynomial time algorithm that given a lattice  $\Lambda \subseteq \mathbb{R}^n$  with  $\det(\Lambda) \leq 1$  finds a non-zero vector  $\mathbf{x} \in \Lambda$  of length  $\|\mathbf{x}\|_\infty \leq \rho$ . Then, for any  $\mathbf{a} \in [0, 1]^n$ , there is a polynomial time algorithm to find  $\mathbf{x} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$  with  $\|\mathbf{x}\|_\infty \leq k$  and so that  $|\langle \mathbf{a}, \mathbf{x} \rangle| \leq 2nk\rho(\frac{\rho}{k})^n$ .

*Proof.* When  $\rho(\frac{\rho}{k})^n \geq \frac{1}{2}$ , we have  $2nk\rho(\frac{\rho}{k})^n \geq 1$ , in which case  $\mathbf{x} = (1, 0, \dots, 0)$  trivially does the job. We can hence assume that  $\rho(\frac{\rho}{k})^n < \frac{1}{2}$ .

Let  $\mathbf{I}_n$  be the  $n \times n$  identity matrix and consider the lattice  $\Lambda$  generated by the  $(n+1) \times (n+1)$ -dimensional matrix

$$B := \begin{pmatrix} \frac{\rho}{k} \mathbf{I}_n & \mathbf{0} \\ \frac{1}{2nk} (\frac{k}{\rho})^n \mathbf{a}^T & (\frac{k}{\rho})^n \end{pmatrix}.$$

Note that  $\det(B) = 1$ . Let  $\mathbf{x} \in \Lambda$  be the vector returned by the algorithm. Then  $\mathbf{y} := B^{-1}\mathbf{x} \in \mathbb{Z}^{n+1}$ . Since  $\|\mathbf{x}\|_\infty \leq \rho$  and  $\mathbf{x} = B\mathbf{y}$ , we have  $|y_i| \leq k$  for  $i = 1, \dots, n$ . Moreover,

$$\begin{aligned} \rho \geq |x_{n+1}| &= \left| y_{n+1} \left( \frac{k}{\rho} \right)^n + \sum_{i=1}^n \frac{1}{2nk} \left( \frac{k}{\rho} \right)^n y_i a_i \right| \geq |y_{n+1}| \left( \frac{k}{\rho} \right)^n - \left| \sum_{i=1}^n \frac{1}{2nk} \left( \frac{k}{\rho} \right)^n y_i a_i \right| \\ &\stackrel{|y_i| \leq k}{\geq} |y_{n+1}| \left( \frac{k}{\rho} \right)^n - \frac{1}{2} \left( \frac{k}{\rho} \right)^n, \end{aligned}$$

which implies  $|y_{n+1}| \leq \frac{1}{2} + \rho(\frac{\rho}{k})^n$ . Since  $\rho(\frac{\rho}{k})^n < \frac{1}{2}$  and  $y_{n+1} \in \mathbb{Z}$ , we must have  $y_{n+1} = 0$ .

Therefore,  $\left| \sum_{i=1}^n \frac{1}{2nk} \left( \frac{k}{\rho} \right)^n y_i a_i \right| = |x_{n+1}| \leq \rho$ , and hence  $|\sum_{i=1}^n y_i a_i| \leq 2nk\rho(\frac{\rho}{k})^n$ , so the vector  $(y_1, \dots, y_n)$  does the job.  $\square$

Now using Lemma 5 and Theorem 8 we are able to prove Theorem 2:

*Proof of Theorem 2.* The proof mirrors that of Theorem 1. We start by choosing  $k = 3\rho$  in Theorem 8, which gives us  $|\langle \mathbf{a}, \mathbf{x} \rangle| \leq 2^{-n}$  and  $\|\mathbf{x}\|_\infty \leq 3\rho$ . As we argued in the proof of Theorem 1, we may assume that  $\rho < \frac{\log n}{48 \log \log n}$ . This gives us  $k \leq \frac{\log n}{16 \log \log n}$  and we can, as before, apply Lemma 5.  $\square$

### 3 Reducing Minkowski's Theorem to Number Balancing

In this section we show how an oracle for number balancing can be used to design an algorithm to approximate Minkowski's problem. More precisely, given a large enough symmetric convex body  $K$ , we will be able to find a vector  $\mathbf{x} \in \mathbb{Z}^n \cap \rho K \setminus \{\mathbf{0}\}$  where  $\rho$  is a polynomial in  $n$ . The first helpful insight is that each such convex body can be approximated within a factor of  $\sqrt{n}$  using an *ellipsoid* [Lov90, MG12]. Recall that an ellipsoid is a set of the form

$$\mathcal{E} = \left\{ \mathbf{x} \in \mathbb{R}^n \mid \sum_{i=1}^n \frac{1}{\lambda_i^2} \cdot \langle \mathbf{x}, \mathbf{a}_i \rangle^2 \leq 1 \right\} \quad (1)$$

with an *orthonormal basis*  $\mathbf{a}_1, \dots, \mathbf{a}_n \in \mathbb{R}^n$  defining the *axes* and positive coefficients  $\lambda_1, \dots, \lambda_n > 0$  that describe the *lengths of the axes*<sup>3</sup>. Overall, our reduction will operate in two steps:

- (i) By combining *John's Theorem* with *lattice basis reduction*, we can show that it suffices to find integer points in an ellipsoid that is *well-rounded*, meaning that the lengths of the axes are bounded.
- (ii) We show that a number balancing oracle allows a self-reduction to a generalized form where inner products with  $n$  vectors have to be minimized and additionally the solution space is  $\mathbb{Z}^n$  instead of  $\{-1, 0, 1\}^n$ .

We begin by proving (ii) and postpone (i) until the end of this section.

#### 3.1 A self-reduction to a generalized form of number balancing

The main technical result of this section is the following reduction:

**Theorem 9.** *Suppose there is a  $\delta$ -approximation for number balancing with error parameter  $\delta = \frac{\rho(n)}{2^n}$  and  $\rho(n) \leq 2^{n/2}$ . Then there is a polynomial time algorithm that on input  $\mathbf{a}_1, \dots, \mathbf{a}_n \in [0, 1]^n$  and  $0 < \lambda_1 \leq \dots \leq \lambda_n \leq 2^n$  with  $\prod_{i=1}^n \lambda_i \geq 1$ , finds a vector  $\mathbf{x} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$  with*

$$|\langle \mathbf{x}, \mathbf{a}_i \rangle| \leq O(n^4) \cdot \lambda_i \cdot \rho(4n^2)^{1/n} \quad \forall i = 1, \dots, n.$$

In particular if  $\rho(n) \leq 2^{\sqrt{n}}$ , then the right hand side in Theorem 9 simplifies to just  $O(n^4) \cdot \lambda_i$ . We will show this by introducing two extensions of the number balancing oracle. The first extension gives a weaker bound in terms of the error parameter, but allows for multiple vectors in  $[0, 1]^n$ . In the second extension, we extend the range of coefficients from  $\{-1, 0, 1\}$  to  $\{-Q, \dots, Q\}$  which leads to a much stronger error bound.

**Lemma 10.** *Suppose there is a  $\delta$ -approximation for number balancing. Then there is a polynomial time algorithm that given an input  $\mathbf{a}_1, \dots, \mathbf{a}_k \in [0, 1]^n$  and parameters  $\delta_1, \dots, \delta_k \leq \frac{1}{2}$  with  $\prod_{i=1}^k \delta_i \geq \delta$  finds a vector  $\mathbf{x} \in \{-1, 0, 1\}^n \setminus \{\mathbf{0}\}$  with  $|\langle \mathbf{a}_i, \mathbf{x} \rangle| \leq 2n^2 \delta_i$  for all  $i = 1, \dots, k$ .*

*Proof.* The idea is that we will discretize all of the vectors and then run our oracle on their sum. The vector that we obtain will then have small inner product with all of the  $\mathbf{a}_i$ . To define the discretization  $\tilde{\mathbf{a}}_i$ , round elements of  $\mathbf{a}_i$  down to the nearest multiple of  $2n\delta_i$ , and then multiply by  $\prod_{j < i} \delta_j$ . Defining  $\tilde{\mathbf{a}}_i$  this way, notice that for all  $i$  we have  $|\langle \tilde{\mathbf{a}}_i, \mathbf{x} \rangle| \leq n \prod_{j < i} \delta_j$ .

<sup>3</sup>Strictly speaking, the length of axis  $i$  is  $2\lambda_i$ , but we will continue calling  $\lambda_i$  the “axis length”.

Now let  $\mathbf{c} = \tilde{\mathbf{a}}_1 + \dots + \tilde{\mathbf{a}}_k$ . By our oracle, we can find  $\mathbf{x} \in \{-1, 0, 1\}^n \setminus \{\mathbf{0}\}$  with  $|\langle \mathbf{c}, \mathbf{x} \rangle| \leq \delta$ . We have

$$\begin{aligned} |\langle \tilde{\mathbf{a}}_1, \mathbf{x} \rangle| &\leq |\langle \tilde{\mathbf{a}}_2, \mathbf{x} \rangle| + \dots + |\langle \tilde{\mathbf{a}}_k, \mathbf{x} \rangle| + |\langle \mathbf{c}, \mathbf{x} \rangle| \leq n\delta_1 + n \prod_{j \leq 2} \delta_j + \dots + n \prod_{j \leq k} \delta_j \\ &\leq n\delta_1 \cdot \left(1 + \frac{1}{2} + \dots + \frac{1}{2^k}\right) < 2n\delta_1. \end{aligned}$$

Therefore  $|\langle \tilde{\mathbf{a}}_1, \mathbf{x} \rangle| = 0$ . Similarly, for  $1 < i \leq k$ , if  $|\langle \tilde{\mathbf{a}}_1, \mathbf{x} \rangle|, \dots, |\langle \tilde{\mathbf{a}}_{i-1}, \mathbf{x} \rangle| = 0$ , then we have

$$|\langle \tilde{\mathbf{a}}_i, \mathbf{x} \rangle| \leq |\langle \tilde{\mathbf{a}}_{i+1}, \mathbf{x} \rangle| + \dots + |\langle \tilde{\mathbf{a}}_k, \mathbf{x} \rangle| + |\langle \mathbf{c}, \mathbf{x} \rangle| < 2n \prod_{j \leq i} \delta_j,$$

and hence  $|\langle \tilde{\mathbf{a}}_i, \mathbf{x} \rangle| = 0$  for all  $i$ . Notice that by definition of  $\tilde{\mathbf{a}}_i$  we have  $\|\prod_{j < i} \delta_j \mathbf{a}_i - \tilde{\mathbf{a}}_i\|_\infty \leq 2n \prod_{j \leq i} \delta_j$ . Therefore  $|\langle \prod_{j < i} \delta_j \mathbf{a}_i, \mathbf{x} \rangle| \leq 2n^2 \prod_{j \leq i} \delta_j$ , and so we can conclude that  $|\langle \mathbf{a}_i, \mathbf{x} \rangle| \leq 2n^2 \delta_i$ .  $\square$

Now we come to a second reduction that takes the oracle constructed in Lemma 10 as a starting point:

**Lemma 11.** *Assume there exists a  $f(n)$ -approximation for number balancing. Let  $\mathbf{a}_1, \dots, \mathbf{a}_k \in [0, 1]^n$  be given with parameters  $\delta_1, \dots, \delta_k \leq \frac{1}{2}$  and a number  $Q$  that is a power of 2 and satisfies  $\prod_{i=1}^k \delta_i \geq f(n \log Q)$ . Then in polynomial time we can find a vector  $\mathbf{x} \in \{-Q, \dots, Q\}^n \setminus \{\mathbf{0}\}$  with  $|\langle \mathbf{a}_i, \mathbf{x} \rangle| \leq \delta_i Q \cdot 2(n \log Q)^2$  for all  $i = 1, \dots, k$ .*

*Proof.* For each  $i$ , we define  $\mathbf{b}_i \in [0, 1]^{n \log Q}$  by  $\mathbf{b}_i(j, \ell) = \mathbf{a}_i(j)2^{-\ell}$  for  $j = 1, \dots, n$  and  $\ell = 1, \dots, \log Q$ . Since  $\prod_{i=1}^k \delta_i \geq f(n \log Q)$ , we can apply Lemma 10 to find  $\mathbf{y} \in \{-1, 0, 1\}^{n \log Q} \setminus \{\mathbf{0}\}$  with  $|\langle \mathbf{b}_i, \mathbf{y} \rangle| \leq \delta_i \cdot 2(n \log Q)^2$ .

Now define  $\mathbf{x} \in \{-Q, \dots, Q\}^n \setminus \{\mathbf{0}\}$  by  $x_j := Q \sum_{\ell=1}^{\log Q} 2^{-\ell} y_{j\ell}$ . Then for  $i = 1, \dots, k$  we have

$$\delta_i \cdot (2n \log Q)^2 \geq |\langle \mathbf{b}_i, \mathbf{y} \rangle| = \left| \sum_{j=1}^n \underbrace{\sum_{\ell=1}^{\log Q} y_{j\ell} 2^{-\ell}}_{=x_j/Q} a_i(j) \right| = \frac{1}{Q} \cdot |\langle \mathbf{a}_i, \mathbf{x} \rangle|$$

and rearranging gives the claim.  $\square$

Finally we come to the proof of Theorem 9.

*Theorem 9.* Suppose that the oracle has parameter  $f(n) = \rho(n)/2^n$ . Suppose that  $\mathbf{a}_1, \dots, \mathbf{a}_n \in [0, 1]^n$  and  $\lambda_1, \dots, \lambda_n > 0$  with  $\prod_{i=1}^n \lambda_i \geq 1$ . We choose  $Q := 2^{4n}$ , which is a power of 2. Define  $\delta_i = \lambda_i \cdot f(n \log Q)^{1/n}$ . Note that  $\delta_i \leq 2^{3n/2} \cdot f(4n^2)^{1/n} \leq \frac{1}{2}$  since  $f(n) \leq 2^{-n/2}$ . Then  $\prod_{i=1}^n \delta_i \geq f(n \log Q)$ , and so by Lemma 11 we can find  $\mathbf{y} \in \{-Q, \dots, Q\}^n \setminus \{\mathbf{0}\}$  with

$$\begin{aligned} |\langle \mathbf{a}_i, \mathbf{y} \rangle| &\leq Q\delta_i \cdot 2(n \log Q)^2 \leq Q\lambda_i \cdot f(n \log Q)^{1/n} \cdot 2(n \log Q)^2 \\ &= \lambda_i \cdot \rho(4n^2)^{1/n} \cdot 2 \cdot (4n^2)^2. \end{aligned}$$

$\square$

### 3.2 A reduction to well-rounded ellipsoids

Using John's Theorem [Joh48], the convex body  $K$  in Theorem 3 can be approximated by an ellipsoid  $\mathcal{E}$  as defined in Eq. (1). The natural approach will then be to apply Theorem 9 to the axes of the ellipsoid. However, it will be crucial that the lengths of the axes of the ellipsoid are bounded by  $2^{O(n)}$ . We will now argue how to make an arbitrary ellipsoid well rounded.

Let us denote  $\lambda_{\max}(\mathcal{E}) := \max\{\lambda_i : i = 1, \dots, n\}$  as the maximum length of an axis. Recall that a matrix  $U \in \mathbb{R}^{n \times n}$  is *unimodular* if  $U \in \mathbb{Z}^{n \times n}$  and  $|\det(U)| = 1$ . In particular, the linear map  $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$



with  $T(\mathbf{x}) = U\mathbf{x}$  is a bijection on the integer lattice, meaning that  $T(\mathbb{Z}^n) = \mathbb{Z}^n$ . It turns out that one can use the *lattice basis reduction* method by Lenstra, Lenstra and Lovász [LLL82] to find a unimodular linear transformation that “regularizes” any given ellipsoid. Note that it suffices to work with the regularized ellipsoid  $T(\mathcal{E})$  since  $\text{vol}_n(T(\mathcal{E})) = \text{vol}_n(\mathcal{E})$  and if we find a point  $\mathbf{x} \in (\rho T(\mathcal{E})) \cap \mathbb{Z}^n$ , then by linearity  $T^{-1}(\mathbf{x}) \in \rho\mathcal{E}$  and  $T^{-1}(\mathbf{x}) \in \mathbb{Z}^n$ .

Given  $\mathbf{b}_1, \dots, \mathbf{b}_n \in \mathbb{R}^n$  we define the *Gram-Schmidt orthogonalization* iteratively as  $\hat{\mathbf{b}}_j = \mathbf{b}_j - \sum_{i < j} \mu_{ij} \hat{\mathbf{b}}_i$ , where  $\mu_{ij} = \frac{\langle \mathbf{b}_j, \hat{\mathbf{b}}_i \rangle}{\|\hat{\mathbf{b}}_i\|_2^2}$ . Notice that we can then write  $\mathbf{b}_j = \hat{\mathbf{b}}_j + \sum_{i < j} \mu_{ij} \hat{\mathbf{b}}_i$ . In particular, suppose  $B$  is the matrix with columns  $\mathbf{b}_1, \dots, \mathbf{b}_n$  and  $\hat{B}$  is the matrix with columns  $\hat{\mathbf{b}}_1, \dots, \hat{\mathbf{b}}_n$ . Then  $B = \hat{B}V$  for an upper triangular matrix  $V$  with ones along the diagonal and  $V_{ij} = \mu_{ij}$  for  $i < j$ .

**Definition 1.** Let  $B \in \mathbb{R}^{n \times n}$  be a lattice basis and let  $\mu_{ij}$  be the coefficients from Gram-Schmidt orthogonalization. The basis is called *LLL reduced* if

- (Coefficient-reduced):  $|\mu_{ij}| \leq \frac{1}{2}$  for all  $1 \leq i < j \leq n$ .
- (Lovász condition):  $\|\hat{\mathbf{b}}_i\|_2^2 \leq 2\|\hat{\mathbf{b}}_{i+1}\|_2^2$  for  $i = 1, \dots, n-1$ .

LLL reduction has been widely used in diverse fields such as integer programming and cryptography [NV10]. One property of the LLL reduced basis is that the eigenvalues of the corresponding matrix  $B$  are bounded away from 0:

**Lemma 12.** Let  $B$  denote the matrix with columns  $\mathbf{b}_1, \dots, \mathbf{b}_n$ . If  $\mathbf{b}_1, \dots, \mathbf{b}_n$  is an LLL-reduced basis with  $\|\mathbf{b}_i\|_2 \geq 1$  for all  $i$ , then  $\|B\mathbf{x}\|_2 \geq 2^{-3n/2} \cdot \|\mathbf{x}\|_2$  for all  $\mathbf{x} \in \mathbb{R}^n$ .

*Proof.* Let  $\hat{B}$  denote the Gram-Schmidt orthogonalization of  $B$ , with columns  $\hat{\mathbf{b}}_1, \dots, \hat{\mathbf{b}}_n$ . Now, for any  $k$ , we can use the properties of LLL reduction to gain the following bound (proposition (1.7) in [LLL82]).

$$1 \leq \|\mathbf{b}_k\|_2^2 = \|\hat{\mathbf{b}}_k\|_2^2 + \sum_{i < k} \mu_{ik}^2 \|\hat{\mathbf{b}}_i\|_2^2 \leq \left(1 + \frac{1}{4} \sum_{i < k} 2^{k-i}\right) \cdot \|\hat{\mathbf{b}}_k\|_2^2 \leq 2^k \cdot \|\hat{\mathbf{b}}_k\|_2^2.$$

In particular,  $\|\hat{\mathbf{b}}_k\|_2^2 \geq 2^{-n}$  for all  $k = 1, \dots, n$ . Now let  $V$  be the matrix so that  $B = \hat{B}V$ , and let  $\mathbf{x} \in \mathbb{R}^n$  with  $\|\mathbf{x}\|_2 = 1$ . Let  $k$  denote the largest index with  $|x_k| \geq 2^{-k}$ . Then

$$|(V\mathbf{x})_k| = \left| x_k + \sum_{j > k} \mu_{kj} x_j \right| \geq |x_k| - \frac{1}{2} \sum_{j > k} |x_j| \geq 2^{-k} - \frac{1}{2} \sum_{j > k} 2^{-j} \geq 2^{-n}.$$

Now, by the orthogonality of  $\hat{\mathbf{b}}_1, \dots, \hat{\mathbf{b}}_n$ , we have

$$\|B\mathbf{x}\|_2^2 = \|\hat{B}V\mathbf{x}\|_2^2 = \sum_{i=1}^n (V\mathbf{x})_i^2 \|\hat{\mathbf{b}}_i\|_2^2 \geq |(V\mathbf{x})_k|^2 \cdot 2^{-n} \geq 2^{-3n}.$$

Taking square roots gives the claim. □

**Lemma 13.** Let  $\mathcal{E} = \{\mathbf{x} \in \mathbb{R}^n : \|A\mathbf{x}\|_2^2 \leq 1\}$  be an ellipsoid. Then in polynomial time, we can find

- (1) either a vector  $\mathbf{x} \in \mathcal{E} \cap \mathbb{Z}^n$
- (2) or a linear transformation  $T$  so that  $T(\mathbf{x}) = U\mathbf{x}$  for a unimodular matrix  $U$  and  $\lambda_{\max}(T(\mathcal{E})) \leq 2^{3n/2}$ .

*Proof.* Use the algorithm of [LLL82] to find a unimodular matrix  $U$  such that  $B = AU$  is LLL reduced. Let  $\mathbf{b}_1, \dots, \mathbf{b}_n$  denote the columns of  $B$ . Notice that if  $\|\mathbf{b}_i\|_2 \leq 1$ , then  $A^{-1}\mathbf{b}_i \in \mathcal{E} \cap \mathbb{Z}^n$ , and so we are done. So assume now that  $\|\mathbf{b}_i\|_2 \geq 1$  for all  $i$ .

Define  $T(\mathbf{x}) = U^{-1}\mathbf{x}$ , and notice that  $T(\mathcal{E}) = \{\mathbf{x} \in \mathbb{R}^n : \|B\mathbf{x}\|_2^2 \leq 1\}$ . We then have

$$\lambda_{\max}(f(\mathcal{E})) = \max_{\mathbf{x} \in f(\mathcal{E})} \|\mathbf{x}\|_2 = \max_{\|B\mathbf{x}\|_2 \leq 1} \|\mathbf{x}\|_2 = \max_{\mathbf{x} \neq 0} \frac{\|\mathbf{x}\|_2}{\|B\mathbf{x}\|_2} \leq 2^{3n/2},$$

where the last inequality follows from Lemma 12.  $\square$

Finally we can prove one of our main results, Theorem 3.

*Theorem 3.* Let  $K \subseteq \mathbb{R}^n$  be a convex body with  $\text{vol}_n(K) \geq 2^n$ . We compute an ellipsoid<sup>4</sup>  $\mathcal{E} = \{\mathbf{x} \in \mathbb{R}^n \mid \sum_{i=1}^n \frac{1}{\lambda_i^2} \langle \mathbf{x}, \mathbf{a}_i \rangle^2 \leq 1\}$  so that  $\frac{1}{5\sqrt{n}}\mathcal{E} \subseteq K \subseteq \frac{1}{5}\sqrt{n}\mathcal{E}$ . Then

$$2^n \cdot 5^n \cdot n^{-n/2} \leq \text{vol}_n(K) \cdot 5^n \cdot n^{-n/2} \leq \text{vol}_n(\mathcal{E}) = \underbrace{\text{vol}(B(\mathbf{0}, 1))}_{\leq 5^n n^{-n/2}} \cdot \prod_{i=1}^n \lambda_i.$$

and hence  $\prod_{i=1}^n \lambda_i \geq 1$ . We apply Lemma 13 to either find an integer point in  $\mathcal{E}$  and we are done, or we find a unimodular transformation  $T$  so that the ellipsoid  $\tilde{\mathcal{E}} := T(\mathcal{E})$  has all axes of length at most  $2^{O(n)}$ . Suppose the latter case happens. We write  $\tilde{\mathcal{E}} = \{\mathbf{x} \in \mathbb{R}^n \mid \sum_{i=1}^n \frac{1}{\tilde{\lambda}_i^2} \langle \mathbf{x}, \tilde{\mathbf{a}}_i \rangle^2 \leq 1\}$  and observe that still  $\prod_{i=1}^n \tilde{\lambda}_i \geq 1$  as the volume of the ellipsoid has not changed. We make use of the  $\delta$ -approximation for the number balancing problem to apply Theorem 9 to the vectors  $\tilde{\mathbf{a}}_1, \dots, \tilde{\mathbf{a}}_n$  and parameters  $\tilde{\lambda}_1, \dots, \tilde{\lambda}_n$  and obtain a vector  $\mathbf{x} \in \mathbb{Z}^n \setminus \{0\}$  with  $|\langle \tilde{\mathbf{a}}_i, \mathbf{x} \rangle| \leq \tilde{\lambda}_i \cdot O(n^4)$ . Then  $\sum_{i=1}^n \frac{1}{\tilde{\lambda}_i^2} \langle \mathbf{a}_i, \mathbf{x} \rangle^2 \leq O(n^9)$  and hence  $\mathbf{x} \in O(n^{4.5}) \cdot \tilde{\mathcal{E}}$ . Then  $T^{-1}(\mathbf{x}) \in (O(n^5) \cdot K) \cap (\mathbb{Z}^n \setminus \{0\})$ .  $\square$

## References

- [Ajt96] M. Ajtai. Generating hard instances of lattice problems. In *Proceedings of the 28th STOC*, pages 99–108. ACM, 1996.
- [AR05] D. Aharonov and O. Regev. Lattice problems in NP cap comp. *J. ACM*, 52(5):749–765, 2005.
- [Boh96] T. Bohman. A sum packing problem of erdős and the conway-guy sequence. *Proceedings of the AMS*, 124(12):3627–3636, 1996.
- [GJ97] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1997.
- [HR07] I. Haviv and O. Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. pages 469–477, 2007.
- [Joh48] F. John. Extremum problems with inequalities as subsidiary conditions. In *Studies and Essays Presented to R. Courant on his 60th Birthday, January 8, 1948*, pages 187–204. Interscience Publishers, Inc., New York, N. Y., 1948.
- [KK82] N. Karmarkar and R. Karp. The differencing method of set partitioning. Technical report, CS Division, UC Berkeley, 1982. <http://digitalassets.lib.berkeley.edu/techreports/ucb/text/CSD-83-113.pdf>.
- [KPP04] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack problems*. Springer, 2004.

---

<sup>4</sup>Note that there *exists* an ellipsoid that approximates  $K$  within a factor of  $\sqrt{n}$  and if  $K$  is a polytope with  $m$  facets, then this ellipsoid can be found in time polynomial in  $n$  and  $m$ . However, if one only has a separation oracle for  $K$ , then the best factor achievable in polynomial time is  $n$ .

- [LLJS90] J. Lagarias, H. Lenstra Jr, and C. Schnorr. Korkin-zolotarev bases and successive minima of a lattice and its reciprocal lattice. *Combinatorica*, 10(4):333–348, 1990.
- [LLL82] A. Lenstra, H. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.
- [Lov90] L. Lovász. *Geometric algorithms and algorithmic geometry*. American Mathematical Society, 1990.
- [Lun88] W. Lunnon. Integer sets with distinct subset-sums. *Mathematics of Computation*, 50(181):297–320, 1988.
- [LY11] V. Lev and R. Yuster. On the size of dissociated bases. *the electronic journal of combinatorics*, 18(1):P117, 2011.
- [Mat02] J. Matousek. *Lectures on Discrete Geometry*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.
- [Mer06] S. Mertens. The easiest hard problem: Number partitioning. *Computational Complexity and Statistical Physics*, 125(2):125–139, 2006.
- [MG12] A. Schrijver M. Grötschel, L. Lovász. *Geometric algorithms and combinatorial optimization*, volume 2, pages 122–125. Springer, 2012.
- [MT90] S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, Inc., New York, NY, USA, 1990.
- [NV10] P. Nguyen and B. Vallée. The lll algorithm. *Information Security and*, 2010.
- [Sch87] C. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.*, 53:201–224, 1987.
- [WY92] G. Woeginger and Z. Yu. On the equal-subset-sum problem. *Information Processing Letters*, 42(6):299–302, 1992.